Computing Exact *p*-values for a Cross-correlation Shotgun Proteomics Score Function

J. Jeffry Howbert‡ and William Stafford Noble‡§¶

The core of every protein mass spectrometry analysis pipeline is a function that assesses the quality of a match between an observed spectrum and a candidate peptide. We describe a procedure for computing exact *p*-values for the oldest and still widely used score function, SEQUEST XCorr. The procedure uses dynamic programming to enumerate efficiently the full distribution of scores for all possible peptides whose masses are close to that of the spectrum precursor mass. Ranking identified spectra by *p*-value rather than XCorr significantly reduces variance because of spectrum-specific effects on the score. In combination with the Percolator postprocessor, the XCorr *p*-value yields more spectrum and peptide identifications at a fixed false discovery rate than Mascot, X!Tandem, Comet, and MS-GF+ across a variety of data sets. ***Molecular & Cellular Proteomics 13: 10.1074/mcp.O113.036327, 2467–2479, 2014.***

A high-throughput proteomics experiment generates many thousands of candidate hypotheses, only a fraction of which are true and an even smaller fraction of which are of significant biological interest. Consequently, the accurate and efficient assignment of statistical confidence estimates to identified fragmentation spectra is critical to making efficient use of shotgun proteomics data sets.

Historically, the field has shifted from focusing on *discrimination*—the ability of a search engine to distinguish between correct and incorrect spectrum identifications—to *calibration*. If a score function is well-calibrated, then the score of *x* assigned to one spectrum is directly comparable to a score of *x* assigned to a different spectrum. A well-known example of poor calibration is the distribution of SEQUEST XCorr scores produced on spectra of varying charges (Fig. 1*A*), such that a score of 1.8 for a doubly charged (2+) spectrum indicates a good quality identification, whereas the same score assigned to a 3+ spectrum corresponds to a much poorer match. Improving the calibration of a given score function across spectra can lead to large improvements in the number of identified spectra at a fixed statistical confidence threshold. Score calibration can be carried out using empirical curve fitting procedures to estimate *p*-values (1, 2, 3) or, more recently, using dynamic programming to calculate an exact *p*-value for each observed score (4). Machine learning post-processors such as PeptideProphet (5) and Percolator (6) simultaneously calibrate scores and incorporate additional information, leading to even larger improvements in identification rates.

In this work, we describe a dynamic programming method for computing exact *p*-values for the oldest and one of the most widely used score functions, SEQUEST XCorr (7, 8). We demonstrate analytically and empirically that the resulting *p*-values are valid relative to a widely accepted null model. Furthermore, we show that, across a variety of data sets, our XCorr *p*-value yields significantly improved statistical power relative to competing, state-of-the-art methods, including SEQUEST, Mascot (9), X!Tandem (10), and Comet (3), and is competitive with other dynamic programming-based calibration methods like MS-GF+ (11). Strikingly, the improved calibration from our scoring scheme is complementary to that provided by Percolator, so that the combination of the two methods yields even better results, evaluated both at the spectrum and peptide levels.

MATERIALS AND METHODS

XCorr Exact p-value Score Function—We use the XCorr score function to generate a score for each peptide-spectrum match (PSM).[1] We then estimate an associated statistical confidence measure (*p*-value) by comparing the observed score to the distribution of scores for all possible peptides having the same precursor mass as the spectrum in the PSM. Computing this distribution by enumerating and scoring all the possible peptides is intractable; for example, there are $\sim 7 \times 10^9$ possible peptides with mass 1000 ± 0.5 Da and 7×10^{21} with mass 2000 ± 0.5 Da. Instead, we use dynamic programming (DP) to directly count the number of peptides occurring at each score in the distribution, without enumeration.

The main requirement for counting peptide scores with DP is that the basic score function is additive (4), *i.e.* a simple sum of products between mass intensities in the observed spectrum and the corresponding putative fragment ions from the matching peptide. XCorr is nearly but not perfectly additive; we describe a slight modification to

From the ‡Department of Genome Sciences, University of Washington, Seattle, Washington; §Department of Computer Science and Engineering, University of Washington, Seattle, Washington

Received, November 18, 2013, and in revised form, May 6, 2014
Published, MCP Papers in Press, June 2, 2014, DOI 10.1074/mcp.O113.036327
Author contributions: J.J.H. and W.S.N. designed research; J.J.H. performed research; J.J.H. contributed new reagents or analytic tools; J.J.H. analyzed data; J.J.H. and W.S.N. wrote the paper.

[1] The abbreviations used are: PSM, peptide-spectrum match; FDR, false discovery rate; DP, dynamic programming.

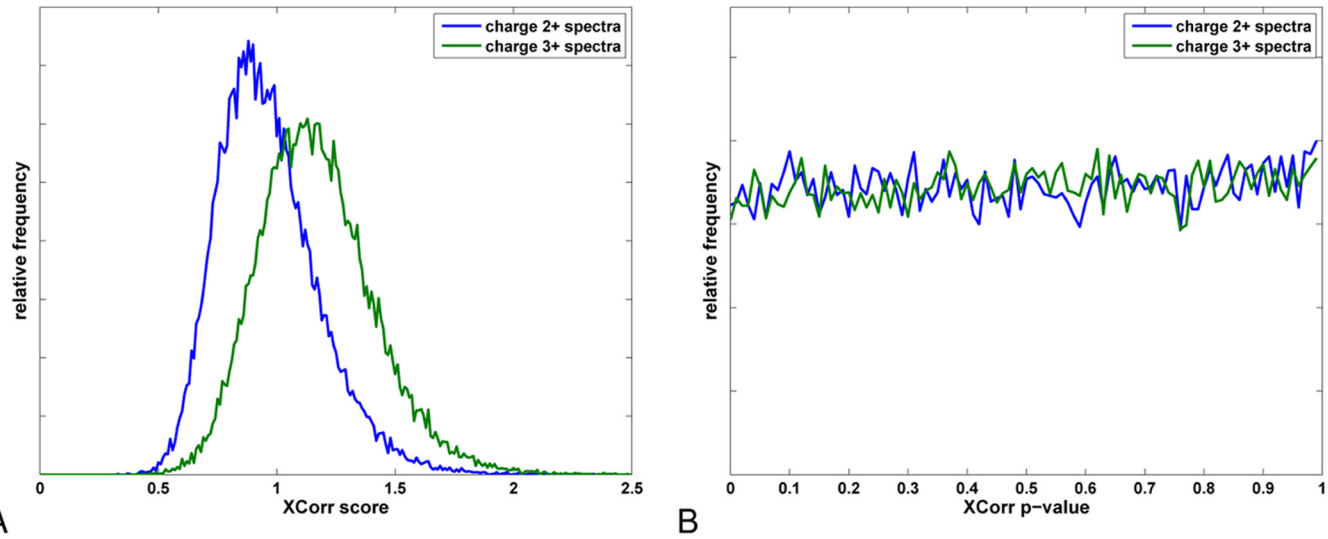

FIG. 1. **Distribution of scores for charge 2+ and charge 3+ spectra from yeast data set.** *A*, Standard XCorr scores. *B*, XCorr Šidák-corrected *p*-values.

make it so. We also show how the computation of XCorr can be refactored to simplify the application of DP.

Standard Computation of XCorr—The form of XCorr preferred in current implementations (3, 8, 12) is the so-called "fast" XCorr (8). Fast XCorr entails four steps of preprocessing the observed spectrum (Fig. 2*A*):

1. The mass axis is discretized by creating a vector *O* of mass bins with bin width = 1.0 Da. Each bin O_i is assigned an intensity value, which is the maximum of the intensities of observed peaks whose masses fall within the mass range of O_i.
2. The intensity of each bin O_i is replaced by its square root.
3. *O* is divided into 10 equal length segments, and the intensities within each segment normalized so the maximum intensity in the segment is 50.
4. A scaled version of *O* is subtracted from itself at each position across a defined window of offsets. This allows the original XCorr's fast Fourier transform calculation of cross-correlation between observed and theoretical spectra to be replaced by a much faster dot product of the preprocessed observed and theoretical spectra.

For the purpose of scoring the match between a peptide and the observed spectrum, XCorr predicts a theoretical spectrum from the sequence of the peptide (Fig. 2*A*):

1. Similar to the observed spectrum, a vector *T* of discrete mass bins (width = 1.0 Da) is created.
2. The masses of the b and y ions resulting from fragmentation of the peptide at each backbone position are predicted, and peaks with uniform intensity are assigned to the corresponding bins of *T*. Note that the original version of XCorr implemented in SEQUEST also added flanking peaks to the theoretical spectrum at ±1.0 Da either side of the b and y ion peaks. We do not include these peaks.
3. The masses of secondary fragmentation products resulting from loss of CO (a-ions), NH_3, and H_2O are predicted and added to *T*. These peaks have a smaller uniform intensity than the primary b and y fragment ions (20% as large).
4. If the precursor charge assigned to the spectrum is >2, then fragments with charge >1 are possible. In this case all the peaks from steps 2 and 3 are replicated into lower-mass bins with the appropriate m/z. In particular, for a precursor of charge *n*, fragment ions up to charge $n - 1$ are considered.
5. The last step deals with the situation where two or more of the peaks predicted in steps 2 - 4 fall in the same bin T_i. In standard fast XCorr, the final predicted intensity in T_i is the maximum of those peaks intensities. However, we wish to make the XCorr score function fully additive, so this step is modified to calculate the predicted intensity in T_i as the sum rather than the maximum of the individual peak intensities.

Once the discretized intensity vectors *O* and *T* have been calculated, the fast XCorr score X_F is computed as a simple dot product between them (Fig. 2*A*).

Refactored Computation of XCorr—Assuming that step 5 in the computation of *T* has been altered to make X_F fully additive, the calculation of XCorr can be refactored as follows (Fig. 2*B*):

1. The preprocessed observed spectrum *O* (from step 4 above) is converted to a new vector *E* with *n* discretized mass bins, where *n* is the integer mass of the spectrum precursor. Each bin E_i specifies the cumulative evidence for cleavage at some hypothetical position on the backbone of the precursor peptide. More precisely, E_i holds the weighted sum of all intensities in *O* whose mass is consistent with a cleavage producing a b ion with integer mass $m_b = i$:

$$E_i = \sum_{m \in I} w_m \cdot O_m \qquad \text{(Eq. 1)}$$

   where *I* are the integer masses of the b, y, and neutral loss ions consistent with m_b and *n*, and w_m are the same relative weights used with X_F for primary *versus* secondary ions in step 3 of computing *T*. As an example, Fig. 2*B* shows the calculation of evidence for hypothetical backbone cleavage at one particular position, where the resulting b ion would have mass 347.

2. The theoretical spectrum *T* used in standard XCorr is replaced by a much simplified discrete mass vector *B*. For each possible backbone fragmentation of the peptide, *B* is populated with a single binary marker at the mass of the corresponding b ion.

The refactored XCorr score X_R is computed as the dot product $E \cdot B$ (Fig. 2*B*).

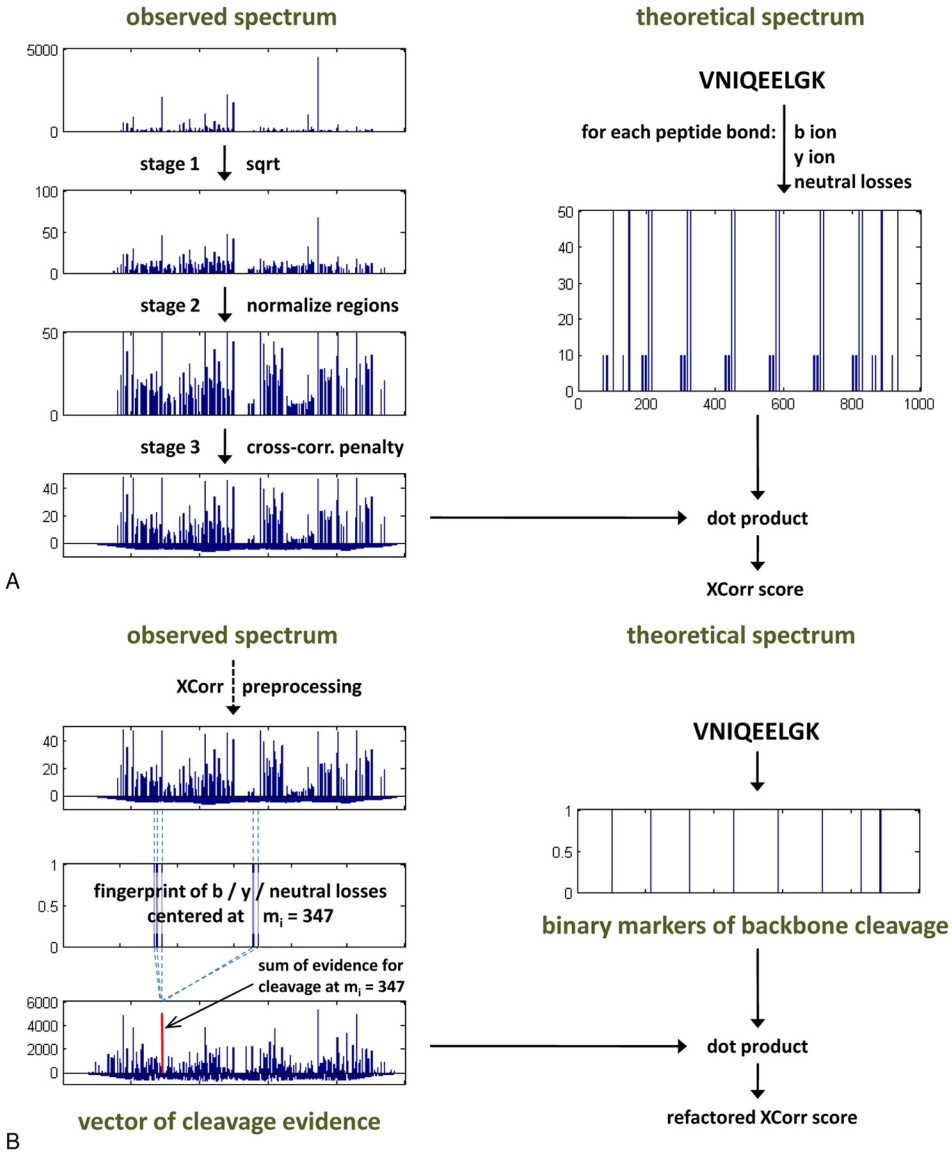

FIG. 2. **XCorr calculation.** *A*, Steps in computing (fast) standard XCorr. *B*, Steps in computing refactored XCorr.

The overall effect of refactoring is to move the detailed specification of potential peptide fragment ions from the theoretical spectrum to the preprocessing of the observed spectrum. Provided that fast XCorr is modified to make it fully additive, we have exactly.

$$X_F = O \cdot T = E \cdot B = X_R \qquad \text{(Eq. 2)}$$

Dynamic Programming to Calculate Score Distribution of all Possible Peptides—The following assumes that a particular spectrum S is being scored, whose precursor mass is m_S.

The final score X_R for a PSM can be expressed as the sum of selected intensities from the evidence vector E, where the elements of B serve as an indicator function to determine which elements E_i are selected:

$$X_R = \sum_{i:B_i = 1} E_i \qquad \text{(Eq. 3)}$$

Let $P^{(1 \to k)}$ be a peptide of length k, mass $m^{(1 \to k)} = m_S$, and amino acid sequence $a_1,a_2, \ldots,a_{k-1},a_k$. Because X_R is additive, the score for matching S with $P^{(1 \to k)}$ can be obtained by first calculating the score for the prefix sequence $P^{(1 \to k-1)} = a_1,a_2, \ldots,a_{k-1}$, then adding the evidence E_{ms} found at mass m_S. Note that this process is equally valid for any subsequence $P^{(1 \to j)} = a_1,a_2, \ldots,a_j$ with mass $m^{(1 \to j)}$,

$$X_R(P^{(1 \to j)}) = X_R(P^{(1 \to j-1)}) + E_{m^{(1 \to j)}} \qquad \text{(Eq. 4)}$$

To calculate the distribution of scores for S using DP, we must discretize the range of scores into uniform bins, as was done with the mass axis. To this purpose, we scale and discretize the values in E so that the largest element takes an integer value of ~100.

Let $C_{s,m}$ be the count of peptides with mass m that produce a discretized score s. If (hypothetically) all the peptides have the same terminal amino acid a with mass m_a, then we would have

$$C_{s,m} = C_{s-E_m,\, m-m_a} \qquad \text{(Eq. 5)}$$

Allowing for all naturally occurring amino acids $a_i \in A$, with masses m_{a_i}, the count becomes

TABLE I
Spectrum data sets used in searches

Data set	Organism	MS1 res	MS2 res	Spectra	Charged spectra	MS1 charges
Yeast	*S. cerevisiae*	Low	Low	35,236	69,705	1+ to 3+
Worm	*C. elegans*	High	Low	7,557	15,871	1+ to 5+
Human-heart	*H. sapiens*	Low	Low	197,987	378,875	1+ to 3+

$$C_{s,m} = \sum_{a_i \in A} C_{s-E_m, m-m_{a_i}} \quad \text{(Eq. 6)}$$

Because X_R is additive, Eq. 6 is valid for all masses $1 \le m \le m_S$. Eq. 6 defines the basic recursion of the DP.

The DP computation of C is conducted in a two-dimensional array, where the rows are indexed by s and the columns by m. The number of rows is determined by estimates of the smallest and largest possible scores for S:

$$s_{min} = \sum_{i=1}^{q} E_{(i)} \quad \text{(Eq. 7)}$$

$$s_{max} = \sum_{i=n-q+1}^{n} E_{(i)} \quad \text{(Eq. 8)}$$

where $E_{(i)}$ refers to the sorted evidence vector values and $q = m_S/min\{m_{a_i \in A}\}$. Note that normally $s_{min} < 0$ because of step 4 in the preprocessing of O.

We initially set:

- $C_{0,1} \leftarrow 1$
- $C_{s,m} \leftarrow 0$ for all $s = 0$ or $m = 1$. This includes a range of indices $s < 1$ and $m < 1$ that are accessed during the DP.

The elements of the array are then computed sequentially:

for $m = 2$ to $m_S - 17$ **do**
  for $s = s_{min}$ to s_{max} **do**
    $C_{s,m} = \sum_{ai \in A} C_{s-E_m, m-m_{a_i}}$
  end for
end for

The last column computed is for mass $m_S - 17$; the final 17 mass units of all peptides are assumed to be a C-terminal -OH group. This column holds the desired distribution of X_R over all possible peptides consistent with m_S.

By using Eq. 6 in the DP, we make the assumption that all peptides are a *priori* equally likely. This is not biologically plausible, and, in fact, leads to distributions of X_R that lack appropriate statistical properties. This problem can be solved by considering the relative abundances of amino acids in the recursive counting:

$$C_{s,m} = \sum_{a_i \in A} C_{s-E_m, m-m_{a_i}} \cdot P_{a_i} \quad \text{(Eq. 9)}$$

where p_{a_i} is the probability of finding amino acid a_i in a large collection of naturally occurring peptides, with $\sum_{a_i \in A} p_{a_i} = 1$. Note that it may be important to use different estimates of p_{a_i} for the N terminus, C terminus, and nonterminal positions, depending on the specificity of the enzyme used for digestion (see RESULTS).

p-values for XCorr Scores—Assume we have calculated, using DP, the distribution of scores C_{s,m_s} over all possible peptides for spectrum S, where $s_{min} \le s \le s_{max}$. Then the p-value relative to this distribution for a specific peptide P, matched to S with XCorr score $X_R' = X_R(P,S)$, is

$$p(X_R', C_{s,m_S}) = \frac{C_{X_R', m_S}/2 + \sum_{s > X_R'} C_{s,m_S}}{\sum_{s_{min} \le s \le s_{max}} C_{s,m_s}} \quad \text{(Eq. 10)}$$

These p-values can be used in place of raw XCorr scores during a standard database search to rank the PSMs for a given spectrum and select the best matching peptide. However, to be able to compare the p-value for the best match $p^{(1)}$ to those for other spectra, it must be corrected for the number n of candidate peptides considered (multiple hypothesis correction). If we assume the data is drawn according to the null distribution and all the n p-values are independent of one another, then the probability we will not observe a p-value $\ge p^{(1)}$ among the set of n p-values is $(1 - p^{(1)})^n$. Accordingly, the corrected p-value p^* for the maximum p-value $p^{(1)}$ among n independent p-values is simply

$$p^* = 1 - (1 - p^{(1)})^n. \quad \text{(Eq. 11)}$$

We refer to p^* as the Šidák-corrected p-value (13). We have consistent empirical evidence that the independence assumption required for the Šidák correction is justified for this type of search (see RESULTS).

False Discovery Rate and q-value Calculations—In general, false discovery rates (FDRs) and q-values are calculated based on separate (*i.e.* nonconcatenated) searches on target and decoy peptide databases, without attempting to account for the presumed proportion π_0 of incorrectly identified target matches ("simple" FDR inference) (14). In brief, the target and decoy p values are combined into a single ranked list. For each target p-value p_i, there are k_{decoy} decoy p-values $\le p_i$ and k_{target} target p-values $\le p_i$, and a FDR of $p_i = k_{decoy}/k_{target}$. It is also feasible to calculate valid q-values using only (Šidák-corrected) p-values for target matches with the Benjamini-Hochberg procedure (15).

Spectrum Data Sets—The three data sets are summarized in Table I, and are available for download at http://noble.gs.washington.edu/proj/exact-pvalue.

Yeast—This spectrum set comprises 35,236 low-resolution MS/MS spectra obtained on a trypsin-digested whole-cell membrane fraction from *S. cerevisiae*, using an LTQ ion trap mass spectrometer. It is more fully described in Reference (6).

Worm—This spectrum set consists of 7557 tandem mass spectra obtained on a trypsin-digested whole-organism lysate of *C. elegans*. MS1 spectra were acquired with high resolution in an Orbitrap mass analyzer, and MS2 spectra with low resolution using an LTQ. It is more fully described in Reference (16). The MS1 and MS2 data were processed with Bullseye/Hardklor (17) to provide high-confidence, high-resolution precursor mass and charge assignments to the MS2 spectra.

Human-heart—This spectrum set is derived from a MudPIT experiment on human heart tissue, as described in Reference (18). We analyzed the first of ten replicate MudPIT experiments on the trypsin-digested healthy human heart tissue sample. Spectrum data was

TABLE II
Peptide data sets used in searches

Data set	Organism	Proteins in original .fasta	Target peptides	Decoy peptides
Yeast	*S. cerevisiae*	6,734	165,930	175,812
Worm	*C. elegans*	23,982	462,511	462,446
Human	*H. sapiens*	70,584	681,367	680,989

obtained with a LTQ linear ion trap mass spectrometer, and comprised 12 .ms2 files, one for each of 12 MudPIT fractions produced in the experiment, which altogether contained 197,987 low resolution MS/MS spectra.

Peptide Data Sets—The three data sets are summarized in Table II, and are available for download at http://noble.gs.washington.edu/proj/exact-pvalue.

Yeast—The original protein .fasta file was downloaded from http://noble.gs.washington.edu/proj/percolator/data/yeast.fasta.gz. Nonredundant target and decoy peptide sets were formed using the create-index function of Crux version 1.38 (12).

Worm—The wormpep194 database was downloaded from http://www.wormbase.org and concatenated with a database of potential human contaminants. Target and decoy peptide sets were created using a custom script, as described in Point 3 under *Search engine comparisons*.

Human—A human reference proteome was downloaded on 2013-01-29 from http://www.uniprot.org/uniprot/?query-organism:9606+keyword:1185&force=yes&format=fasta. Target and decoy peptide sets were created using a custom script, as described in Point 3 under *Search engine comparisons*.

Search Engine Comparisons—Great care was taken to ensure a fair comparison of results across all search engines.

1. Experimental parameters were matched as exactly as possible. For simplicity, all searches were run with full tryptic digestion (*i.e.* no missed cleavages), no nontryptic termini, and no allowance for isotope errors. The only amino acid modification allowed was a fixed carbamidomethyl modification to cysteine. The tolerance window for selecting candidate peptides from the databases was ±3 Da for MS1 low-resolution data (yeast, human-heart) and ±10 ppm for MS1 high-resolution data (worm). These choices of experimental parameters are likely not optimal for some or all of the search engines, but provide a fixed and easily understood basis for comparison.
2. Protein .fasta files were predigested *in silico*. The resulting peptides were placed in new .fasta files that were used for the target searches. All digestions used a trypsin rule without restriction on the C-terminal amino acid (*i.e.* no proline restriction: [KR]|[X]).
3. For every target peptide set, a corresponding decoy peptide set was generated by randomly shuffling the non-C-terminal residues of each target peptide until (if possible) a novel sequence was obtained. After the shuffling process, decoy peptides that were identical to any peptide in the target set or another peptide in the decoy set were removed. The size ratio of target to decoy peptide sets was controlled to be ~1.0.
4. Spectrum files (.ms2 and .mgf) were modified so that every search engine was forced to find matches to an identical set of spectrum-charge combinations.
5. Searches were conducted on nonconcatenated peptide sets: target and decoy peptide sets were searched separately. Score function performance was compared via absolute ranking (*q*-value) curves, generated via a target/decoy analysis appropriate for nonconcatenated searches (14).

The following publicly available versions of search engine software were used.

- XCorr scores were computed with the search-for-matches function of Crux version 1.39 (12).
- MS-GF+: version 20130103 (11).
- Comet: version 2012.01 rev. 3 (3).
- X!Tandem: version 2012.10.01.1 (10).
- Mascot: version 2.3.01; run on Mascot server installed at the University of Washington Proteomics Resource (9).
- Percolator: as implemented in Crux version 1.40 (12).
- MS-GF+ plus Percolator: MS-GF+ version 20140210 and Percolator version 2.06 (19).

The code used for calculating XCorr *p*-values in these studies was written in a mixture of MATLAB and C++. The dynamic programming and other compute-intensive routines were fully coded in C++ for speed.

RESULTS

Reproduction of Original XCorr Score by Refactored XCorr—The XCorr score function compares an observed fragmentation spectrum to a theoretical spectrum derived from a given candidate peptide. Scoring can be carried out as a simple scalar product between the theoretical spectrum and a suitably pre-processed version of the observed spectrum (8). To enable dynamic programming relative to XCorr, we perform an additional pre-processing step on the observed spectrum that summarizes, for each potential b ion, intensity information from the locations of the corresponding y ion and neutral loss peaks. The theoretical spectrum can then be simplified to a Boolean vector with "True" entries indicating the location of peptide b ions (Fig. 2*B*). After these changes, and after discretizing the peak intensity values, the refactored XCorr score function almost exactly reproduces the original XCorr scores: for a randomly chosen representative spectrum from the yeast data set, the Pearson correlation between the original XCorr score and our modified version is 0.995 (Fig. 3*A*).

Statistical Validation of XCorr p-values—Having thus transformed XCorr, we use dynamic programming to compute, for any observed spectrum, the exact distribution of scores relative to all possible peptide sequences whose mass matches the precursor mass of the spectrum (see METHODS AND MATERIALS). Comparison of the XCorr score for a given peptide-spectrum match (PSM) to this distribution produces, by construction, a *p*-value relative to the null hypothesis that the peptide is drawn randomly from all possible sequences.

To test the correctness of our implementation as well as the plausibility of the selected null model, we performed two validation experiments. First, we generated decoy peptides according to the null model, by shuffling the nonterminal amino acids of peptides in our yeast database, and computed 10,000 PSMs using the set of yeast spectra. A quantile-quantile plot (Fig. 3*B*) and a Kolmogorov-Smirnov test ($p = 0.14$) confirm that the resulting *p*-values are uniform. To achieve the desired uniform distribution of *p*-values, it is important to use different estimates of amino acid abundances

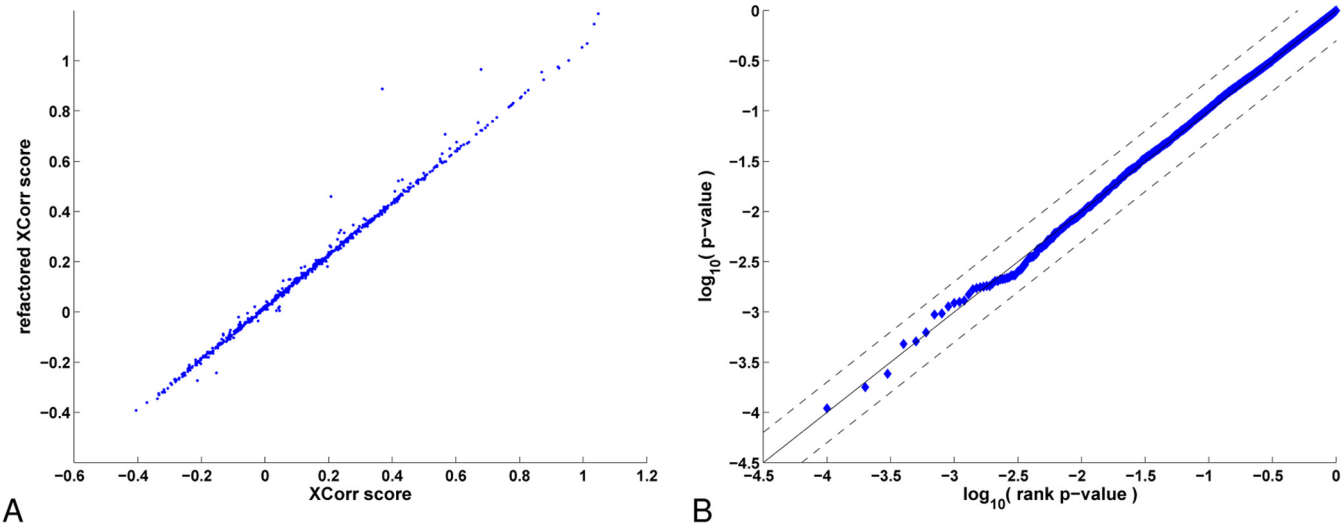

FIG. 3. **Refactored XCorr.** *A*, Correlation between refactored and original XCorr scores for 668 target peptides matched against spectrum 4078 (charge 2+) from yeast data set ($\rho = 0.995$). *B*, Estimated XCorr *p*-values plotted (log-log) as a function of *p*-value rank for 10,000 random decoy PSMs from the yeast data set. Deviation from the solid diagonal line indicates divergence from an ideal null distribution of *p*-values. The dashed diagonal lines ($y = 2x$ and $y = 0.5x$) delimit the range of 2-fold deviation between calculated and ideal *p*-values. The plotted distribution of *p*-values is not different from a uniform distribution on the interval [0,1] by a Kolmogorov-Smirnov test ($p = 0.14$).

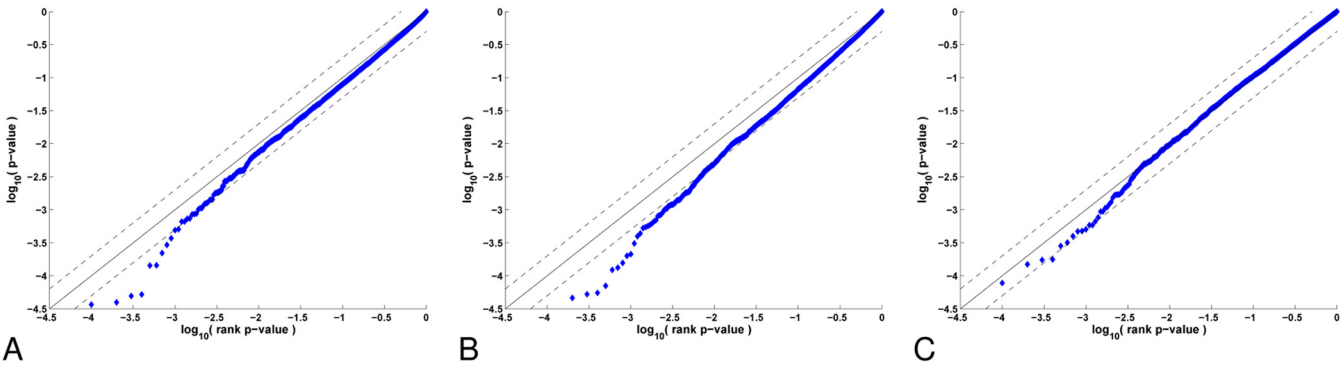

FIG. 4. **Effect of incorporating amino acid abundances into dynamic programming.** For 10,000 random decoy PSMs from the yeast data set, estimated XCorr *p*-values are plotted (log-log) as a function of *p*-value rank. Deviation from the solid diagonal line indicates divergence from an ideal null distribution of *p*-values. The dashed diagonal lines ($y = 2x$ and $y = 0.5x$) delimit the range of 2-fold deviation between calculated and ideal *p*-values. *A*, No incorporation of amino acid abundances (Eq. 6). *B*, Incorporation of amino acid abundances (Eq. 9) using identical abundance estimates at all peptide positions. *C*, Incorporation of amino acid abundances (Eq. 9) using different position-specific abundance estimates for N-terminal, C-terminal, and nonterminal positions. For *B*, and *C,* abundance estimates were calculated empirically from the decoy peptide database.

for the N terminus, C terminus, and nonterminal positions, as determined by the specificity of the enzyme used for digestion (Fig. 4).

Second, we tested the validity of the *p*-values in the context of a database search. After the search, the *p*-values of the top-ranking target and decoy PSMs for each spectrum-charge combination were subjected to Šidák correction. A quantile-quantile plot shows the corrected top-ranking decoy *p*-values conform to the desired null distribution (Fig. 5*A*). We also compared two different methods for calculating FDRs from the corrected *p*-values: a standard Benjamini-Hochberg estimation (15) using only target *p*-values, and a target/decoy estimation procedure in which the corrected *p*-values are essentially treated as scores (14) (see MATERIALS AND METHODS). A plot of the number of accepted PSMs as a function of FDR threshold shows the two estimation procedures yield very similar FDR estimates (Fig. 5*B*), supporting the idea that the Šidák-corrected *p*-values are statistically valid and can be used directly to estimate FDRs without requiring a decoy database search. We found further evidence that the XCorr *p*-value improves score calibration by visualization of target *versus* decoy scores (Fig. 6) and comparison of charge 2+ and 3+ score distributions (Fig. 1).

Comparison of XCorr p-value Performance with Other Score Functions—Having established the validity of our *p*-values, we proceeded to compare the performance of the XCorr *p*-value to several existing database search engines, using three collections of mass spectra: 35,236 fragmentation

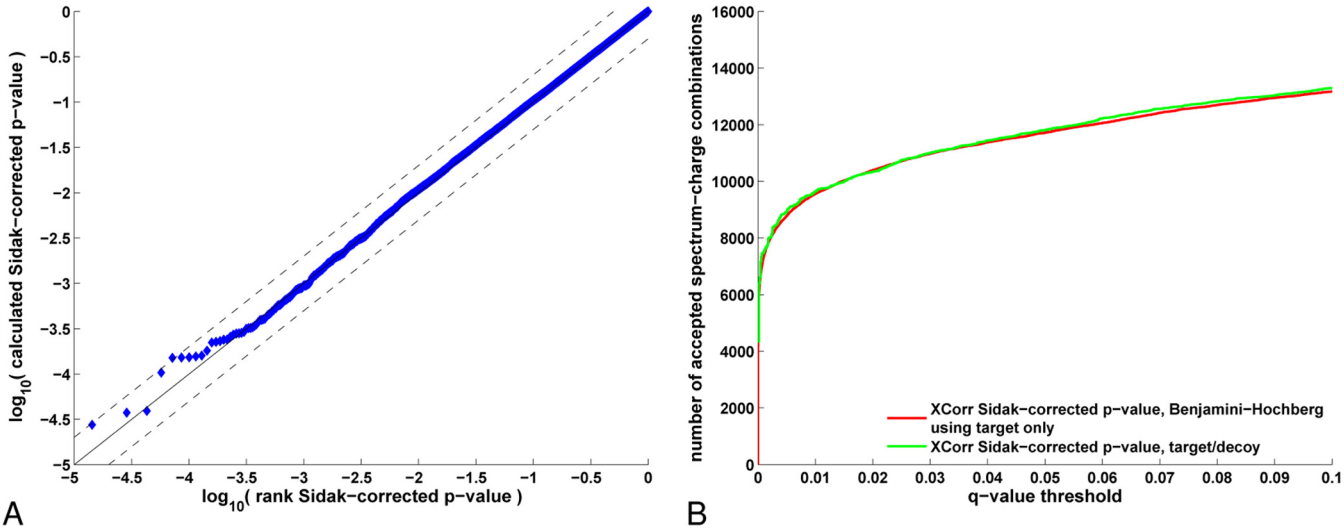

FIG. 5. **Calibration after Šidák correction.** The collection of 69,705 spectrum-charge combinations in the yeast data set were searched against a target and corresponding decoy database. The best *p*-value for each PSM was corrected for the multiple hypotheses tested (number of candidate peptides) using the Šidák correction. *A*, Distribution of decoy XCorr Šidák-corrected *p*-values plotted (log-log) as a function of *p*-value rank. Deviation from the solid diagonal line indicates divergence from an ideal null distribution of *p*-values. The dashed diagonal lines ($y = 2x$ and $y = 0.5x$) delimit the range of 2-fold deviation between calculated and ideal *p*-values. *B*, Number of spectrum-charge combinations accepted as a function of *q*-value threshold. The two series correspond to the same set of PSMs but two different methods for estimating the FDR.

spectra collected from a yeast whole-cell lysate using an LTQ ion trap with low-resolution MS1 (yeast); 7557 fragmentation spectra collected from a *C. elegans* whole-organism lysate, with high-resolution MS1 spectra measured in an Orbitrap, and low-resolution MS2 in an LTQ (worm); and 197,987 fragmentation spectra collected in a 12-step MudPIT experiment from a sample of human heart, using an LTQ linear ion trap with low-resolution MS1 (human-heart). In these experiments, to permit a fair comparison among database search methods, we used a decoy-based confidence estimation procedure (14). We also went to great lengths to ensure that each search engine used comparable parameters and examined exactly the same sets of charge states and target and decoy peptides (see METHODS AND MATERIALS for additional details and references).

For all three data sets we observed a dramatic improvement in statistical power from XCorr *p*-value relative to raw XCorr (Fig. 7): at $q \leq 0.01$, we identify 1885, 337, and 1887 more spectrum-charge combinations for the yeast, worm and human-heart data sets, respectively. Given the calibration results shown above, this level of improvement is not surprising. Indeed, similar effects have been demonstrated previously by performing parametric calibration of X!Tandem (1) or XCorr (2, 3), or by comparing the raw MS-GF score to the corresponding spectral probability (20).

We then compared the performance of the XCorr *p*-value to scores produced by a variety of other search engines (Fig. 7). Across all three data sets and all *q*-value thresholds up to 0.1, XCorr *p*-value outperforms the Mascot ion score and X!Tandem E-value, and outperforms the Comet E-value on the yeast and worm data sets. The spectral E-value produced by MS-GF+ and XCorr *p*-value are competitive over all $q \leq 0.1$ on all three data sets. When we repeat our analysis at the peptide level, using the WOTE method to combine top spectrum scores into top scores for unique peptides (21), the relative performance of the various methods is generally maintained (Fig. 8).

Postsearch Processing of XCorr p-values—In practice, most analysis pipelines do not use search engine scores directly but postprocess them using an algorithm such as Percolator or PeptideProphet. We therefore modified Percolator to take as input the log-transformed XCorr *p*-value in place of the raw XCorr score. For comparison, we also post-processed MS-GF+ scores with Percolator, as recently described (19). At the spectrum level, XCorr *p*-value plus Percolator outperformed all other methods on the yeast and human-heart data sets, including Percolator applied to raw XCorr or MS-GF+ scores, at all *q*-value thresholds we considered (Fig. 7). On the worm data set, however, postprocessing with Percolator showed little gain over XCorr *p*-value, whereas its combination with MS-GF+ provided some improvement, and was the best overall method on this data set. At the peptide level, XCorr *p*-value plus Percolator and MS-GF+ plus Percolator had similar performance on all data sets (Fig. 8).

Identification of Chimeric Spectra—By making a small modification to the method for calculating FDRs, we could also use XCorr *p*-values to identify potential chimeric spectra. Most shotgun proteomics analysis pipelines map each observed spectrum to a single top-scoring candidate peptide from the database. In practice, however, co-eluting isobaric

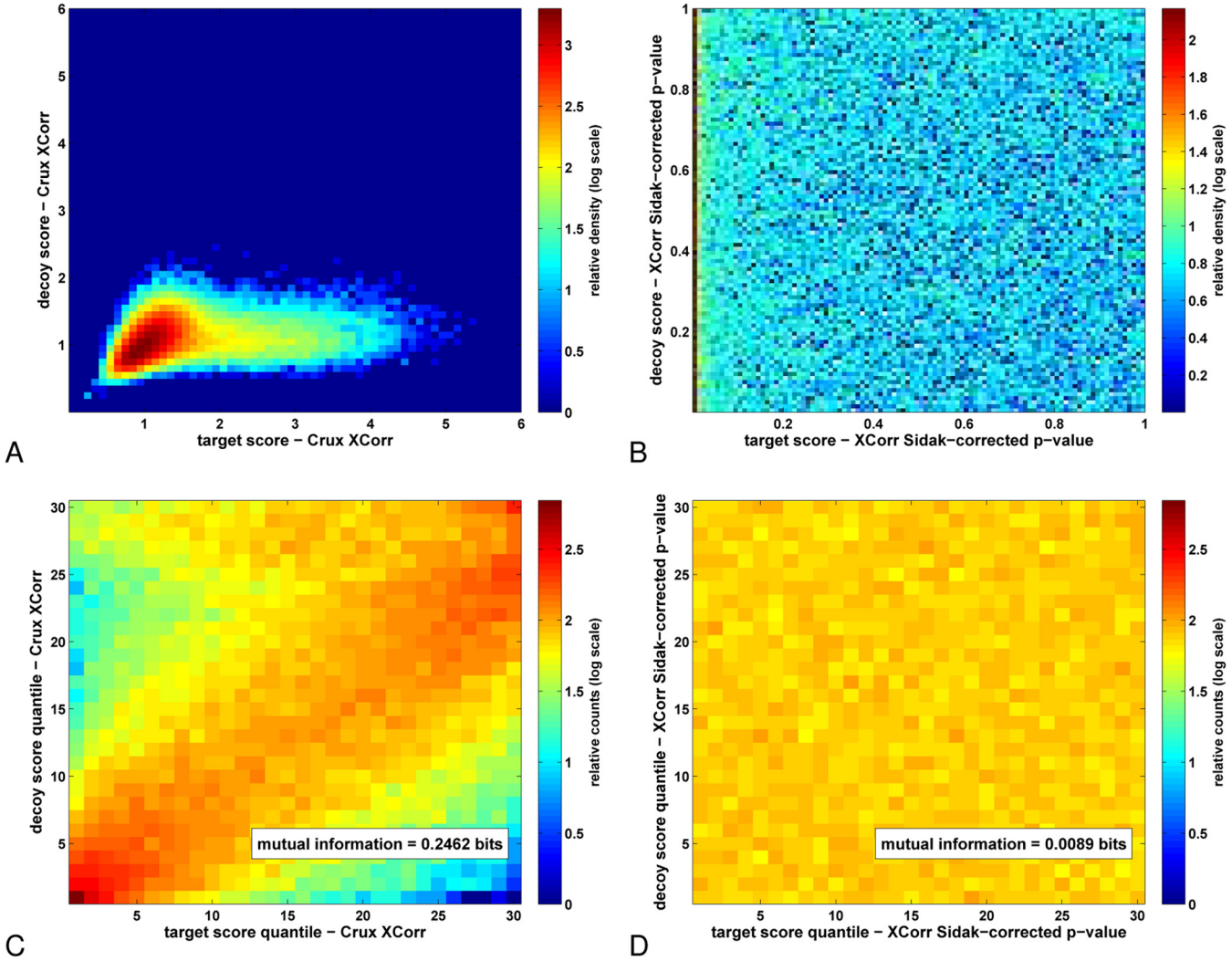

FIG. 6. **XCorr before and after calibration.** The collection of 69,705 spectrum-charge combinations in the yeast data set were searched against a target and corresponding decoy database. *A,* Density plot of target *versus* decoy XCorr scores. Colors indicate the relative density of points (log scale) in a scatter plot of target *versus* decoy values. *B*, Similar to *A*, but using XCorr Šidák-corrected *p*-values instead of XCorr scores. The diagonal component in panel *A* indicates a strong dependence between target and decoy scores ($\rho = 0.29$), whereas the plot in panel *B*, shows a nearly complete absence of correlation ($\rho = 0.014$). *C*, Mutual information of target and decoy XCorr scores. The range of target values was divided into 30 quantiles, and likewise for the range of decoy values. These quantiles defined a 30×30 grid of bins into which target-decoy value pairs were counted. Because of the quantile-quantile binning, the expected value of counts in each bin is uniform if the two covariates are independent. Colors indicate the relative counts of values (log scale) falling in each bin. *D*, Similar to *C*, but using XCorr Šidák-corrected *p*-values instead of XCorr scores. Note that the range of counts represented on the colorbar is the same for plots *C* and *D*. In panel *C* the target and decoy XCorr scores are clearly dependent, whereas panel *D* shows the XCorr *p*-values are independent; the mutual information is in the range found for 10 random shufflings of the target and decoy *p*-values (0.0087 ± 0.0004 bits).

peptides may be simultaneously fragmented in the mass spectrometer, yielding a chimeric spectrum containing peaks from multiple peptides. Having an accurate *p*-value for all candidate PSMs allows us to directly identify such chimeras. We applied Benjamini-Hochberg FDR estimation (15) directly to the complete collection of 25,410,165 target PSMs derived from searching the yeast spectra against the yeast target database. This procedure identified 9989 PSMs with $q \leq 0.01$. As a control, we applied the same analysis to the 26,893,412 decoy PSMs derived from searching against the yeast decoy database; this procedure failed to identify any PSMs with $q \leq 0.01$. When compared with the traditional approach of computing FDRs using only the top target and decoy match for each spectrum, this method of FDR estimation identifies significantly more PSMs because of the presence of chimeric spectra (Fig. 9*A*). Among the 9989 PSMs identified from the target search, there were 9707 unique spectra, of which 9431 spectra were matched to a single peptide, 270 spectra were matched to two peptides, and six spectra were matched to three peptides. To distinguish between true chimeric spectra and spectra that match two similar peptide sequences, we converted leucine to isoleucine and then applied two filters to

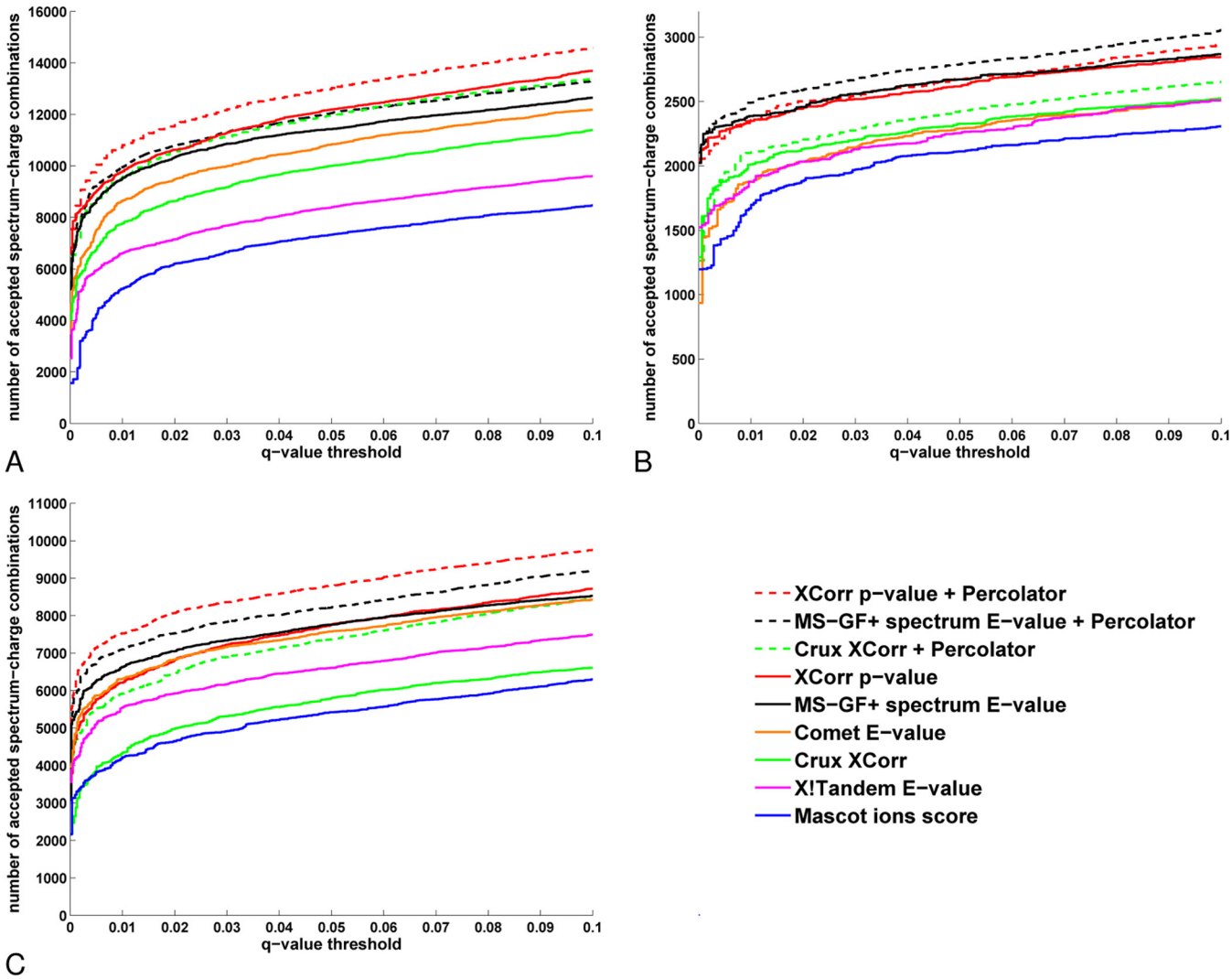

FIG. 7. **Comparison of search tools - spectrum identification.** Each series plots, for a given score function, the number of accepted spectrum-charge combinations from the indicated data set as a function of *q*-value threshold. *A*, Yeast data set. *B*, Worm data set. *C*, Human-heart data set. All *q*-values are estimated using a decoy database estimation procedure (14).

eliminate chimeric matches to pairs of peptides with similar sequences. Out of 270 spectra matching two peptides, 193 passed both filters: normalized edit distance greater than 0.35 between the peptides, and fraction of shared b and y ion peaks less than 0.40. We compared these 193 chimeras to the 170 chimeras with nonsimilar peptides found by applying Percolator with the same criteria to the top 5 raw XCorr PSMs for each spectrum (6). These chimeras only partially overlapped with those identified by Percolator. A total of 101 chimeras were found in both sets, whereas 92 chimeras were found only using XCorr *p*-values and 69 chimeras were found only with XCorr plus Percolator.

To further validate the chimeras discovered using XCorr *p*-values, we chose a representative chimeric spectrum for which authentic library spectra exist in the NIST collection for both peptides. Alignment of the putative chimeric spectrum with the library spectra (Fig. 9*B*) shows excellent correspondence between dominant intensities at the predicted b and y ion masses in the library spectra and the dominant intensities in the chimeric spectrum. The correlation between the b/y ion intensities for peptide 1 (TAGIQIVADDLTVTNPAR) and the intensities at the same masses in the chimeric spectrum is $\rho = 0.9042$, and for peptide 2 (EYIFSENSGVLGDVAAGK) it is $\rho = 0.8341$. Overall, these results suggest that the XCorr *p*-value is capable of identifying chimeric spectra. Note, however, that separately scoring each peptide in a chimeric spectrum, as we have done, requires that both peptides yield peaks with similar overall intensities. Joint identification of more difficult chimeras, with peptides whose fragment ions exhibit significantly different intensities, will likely require custom score functions and search algorithms (22, 23).

Computational Runtime—The overall runtime of database searches using our XCorr *p*-values was slower than for other search engines, except for MS-GF+, where they were com-

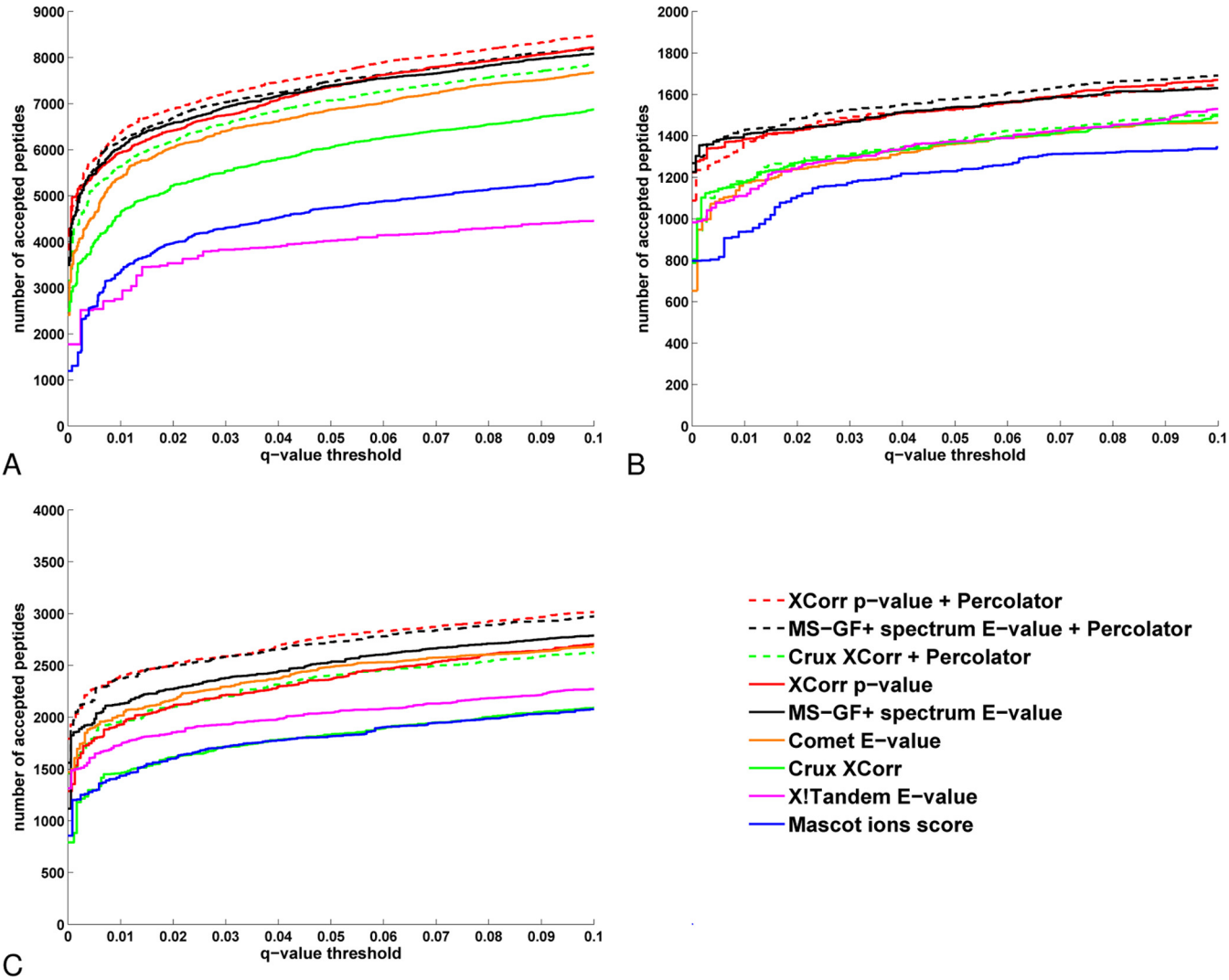

FIG. 8. **Comparison of search tools - peptide identification.** Each series plots, for a given score function, the number of accepted unique peptides from the indicated data set as a function of *q*-value threshold. *A*, Yeast data set. *B*, Worm data set. *C*, Human-heart data set. All *q*-values are estimated using a decoy database estimation procedure (14).

parable. MS-GF+ also computes scores using dynamic programming. A search of the 69,705 charge-spectrum combinations in the yeast data set with our software took 3.8 h on a Linux desktop system, while MS-GF+ required 4.4 h on the same system (combined time for two separate searches on the target and decoy peptide sets). For a search of the 15,871 charge-spectrum combinations in the worm data set, our software took 8.7 min on a Windows laptop, while MS-GF+ required 22.2 min (combined time) on the same system.

DISCUSSION

We have described a method for calculating *p*-values for the widely used XCorr score function. Our method does not rely on parametric estimation (1, 2, 3) or on comparison to an empirical distribution of decoy scores. Instead, *p*-values are generated by a calculation over the set of all possible peptides whose masses match the MS1 precursor. Relative to this null distribution of scores, the *p*-values are exact by construction and display appropriate statistical properties in multiple empirical tests. This approach to calculating XCorr *p*-values is independent of the precursor resolution associated with observed spectra and the size of the database being searched. The method leads to improved performance compared with raw XCorr scores and a parametric method for estimating XCorr *p*-values (3), as well as several other tandem mass spectrometry search engines.

In essence, our approach is an application of the so-called "generating function" framework proposed by Kim *et al.* (4) to a well-known and easy-to-understand score function, the SEQUEST XCorr. To make our exposition self-contained, we have opted to present here our complete model along with the corresponding dynamic programming algorithm, rather than only highlighting differences relative to MS-GF. In practice, the dynamic programming recurrences outlined here are sim-

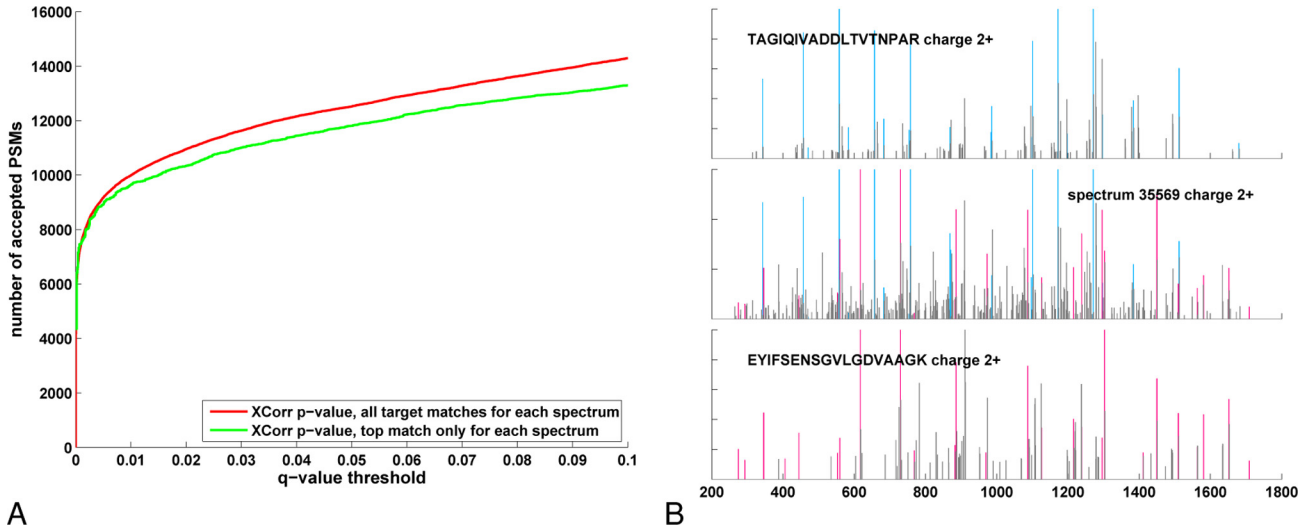

FIG. 9. **Identification of chimeric spectra.** *A*, The green line includes only the top scoring target match for each spectrum; the raw *p*-values were Šidák corrected for multiple testing and *q*-values computed via the target/decoy method (14). The red line includes all target matches for each spectrum; *q*-values were calculated directly from raw *p*-values by the Benjamini-Hochberg method (15). The gap between the two curves indicates the number of chimeric spectra identified at a given *q*-value. *B*, The selected spectrum (center) is aligned with library spectra (top and bottom) for the two peptides identified in the chimera. Colored peaks indicate the predicted masses of b and y ions for each peptide.

ilar to those used in MS-GF, though of course our particular choice of score function is different. We have also omitted Kim *et al.*'s analogies to the Ising model of statistical mechanics and the corresponding energy functions, choosing to describe instead the exact enumeration of our null model in a hypothesis testing framework.

The *p*-value method improves the calibration of XCorr scores to an extent similar to that afforded by Percolator. Furthermore, we observe a strong synergy between the XCorr *p*-value and Percolator on two of the three data sets studied (yeast, human). Percolator is a postprocessing algorithm that builds a semi-supervised model for discriminating correct from incorrect matches (6). It uses information extrinsic to the MS2 fragmentation spectrum, such as the length of the peptide, the number of tryptic termini, the difference between the masses of the MS1 precursor and the peptide in the PSM, and precursor charge. In contrast, calculation of our XCorr *p*-values depends almost entirely on information intrinsic to MS2 spectrum, namely fragment masses and intensities, along with the mass and b ion pattern of the matched peptide. Because the two sources of information are distinct, we believe they may lead to calibration by fundamentally different paths, and the synergy between them is therefore not entirely surprising.

The *p*-value calculated by this method for a particular PSM depends only on the spectrum and peptide involved in that PSM, and is independent of the PSMs for any other peptides in the target and decoy databases. Thus a database search over a set of spectra can generate a meaningful *p*-value for each spectrum by searching only against the target peptides, and a parallel search against decoy peptides is, in principle, superfluous. As shown in the RESULTS, FDR analysis using target peptide *p*-values alone leads to essentially the same number of accepted spectra as an FDR analysis that combines target and decoy *p*-values. However, unlike most search engines, the elimination of decoy searching from our method does not lead to a possible twofold savings in search time (see below). In addition, there are situations where results from decoy searches are required, *e.g.* postprocessing of scores using PeptideProphet or Percolator.

It should be straightforward to generalize our method to more complex tandem mass spectrometry experiments. For example, post-translational modifications are easily accommodated by extending the set of amino acids that are summed over in the dynamic programming. One needs to address, however, the issue of estimating the abundances of modified amino acids needed for Eq. 9. Application to experiments involving cross-linked peptides also appears feasible, although the dynamic programming will necessarily be considerably more complex.

With a simple change of bookkeeping, database searches can be modified to return the best *p*-value obtained for each peptide in the database, rather than the best *p*-value for each spectrum. We have observed that the resulting *p*-values, after Šidák correction, are not uniformly distributed (data not shown), presumably because multiple spectra generated by the same peptide ion species cannot be considered statistically independent. Nonetheless, the resulting peptide "*p*-values" can be expected to produce better peptide rankings than raw XCorr scores. This opens the possibility of conducting searches in a "peptide-centric" manner, rather than the traditional "spectrum-centric" approach. Such an

approach would be highly desirable in experimental settings such as data-independent acquisition, where spectra are expected to routinely contain fragments from multiple precursors.

The dynamic programming that lies at the core of our *p*-value calculation is computationally intensive, consuming around 85%-90% of the overall runtime. Its computational complexity, in formal terms, is $O(|A|\cdot(s_{max} - s_{min})\cdot m_S)$, *i.e.* the time required for dynamic programming scales linearly in each of three factors: $|A|$, the number of amino acids; $(s_{max} - s_{min})$, the number of possible discretized scores; and m_S, the presumed mass of the MS1 precursor of the MS2 fragmentation spectrum. For the most part, these factors are inherent to the matrix over which dynamic programming is performed and not amenable to modification. The exception is $(s_{max} - s_{min})$, which can be controlled by the choice of bin size when discretizing the evidence vector. In our current implementation, we use the largest bin size that does not degrade FDR performance. In the future, we plan to investigate carrying out the dynamic programming using a graphics processing unit, which should significantly speed up the calculation.

Generation of an evidence vector and the associated dynamic programming require a choice of value for the discretized mass of the spectrum precursor. During database search, this value needs to range over the discretized masses of all candidate peptides selected from the database for that spectrum. In consequence, the computational expense of dynamic programming is also related to the size of the mass window used to select peptide candidates. For typical searches on spectra with low-resolution MS1 using, for example, a mass selection window of ±3.0 Da, six or seven unique discretized masses will occur among the candidate peptides, and dynamic programming must be run separately for each mass. For spectra with high-resolution MS1, only a single unique discretized mass will occur among the candidate peptides. In any event, the same dynamic programming-derived score distributions are used to calculate *p*-values for both target and decoy peptides, so that running a decoy search in addition to target search adds only marginally to overall runtimes.

We have observed that the score distributions derived via dynamic programming have similar shapes but different widths, depending on the mass chosen for the spectrum precursor. One direction for our future investigations will explore the generation of distributions using a parametric model, where the structure of the model is based on a small number of canonical distributions obtained from dynamic programming.

The software for doing database searches with our XCorr *p*-values is being integrated into the freely available Crux software toolkit (http://noble.gs.washington.edu/proj/crux). The current beta version of the integration, including source code, can be downloaded from http://noble.gs.washington.edu/proj/exact-pvalue. Details on obtaining the data used in this manuscript may be found in MATERIALS AND METHODS.

In this study, the best overall statistical performances were clearly produced by the dynamic programming-based score functions, MS-GF+ and our XCorr *p*-values. We see advantages to each method. MS-GF+ is the latest advance in a sequence of search tools going back several years (4), and has a mature set of features, including the ability to search on high-resolution MS2 data and variable modifications. Comparable features for our method are currently under development. However, whereas the underlying score function in MS-GF+ is novel, we have built upon the widely used and well understood XCorr score function.

After submission of this manuscript, we became aware of a previous report in which statistical confidence measures were estimated for the XCorr score function using dynamic programming (24). Although similar to our method, the reproduction of the XCorr score by (24) is only approximate, whereas ours is nearly exact (Fig. 3*A*). We also note that, unlike MS-GF+ and the method of (24), our code is open source, making the details of our method transparent and reproducible, as well as modifiable by other investigators.

¶ To whom correspondence should be addressed: Box 355065, Foege Building, S220B, 3720 15th Ave NE, Seattle, WA 98195-5065. E-mail: william-noble@uw.edu.

REFERENCES

1. Fenyo, D. and Beavis, R. C. (2003) A method for assessing the statistical significance of mass spectrometry-based protein identification using general scoring schemes. *Anal. Chem.* **75,** 768–774.
2. Klammer, A. A., Park, C. Y. and Noble, W. S. (2009) Statistical calibration of the SEQUEST XCorr function. *J. Proteome Res.* **8,** 2106–2113.
3. Eng, J. K., Jahan, T. A. and Hoopmann, M. R. (2012) Comet: an open source tandem mass spectrometry sequence database search tool. *Proteomics* **13,** 22–24.
4. Kim, S., Gupta, N. and Pevzner, P. A. (2008) Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J. Proteome Res.* **7,** 3354–3363.
5. Keller, A., Nesvizhskii, A. I., Kolker, E. and Aebersold, R. (2002) Empirical statistical model to estimate the accuracy of peptide identification made by MS/MS and database search. *Anal. Chem.* **74,** 5383–5392.
6. Käll, L., Canterbury, J., Weston, J., Noble, W. S. and MacCoss, M. J. (2007) A semi-supervised machine learning technique for peptide identification from shotgun proteomics datasets. *Nat. Methods* **4,** 923–25.
7. Eng, J. K., McCormack, A. L. and Yates, III, J. R. (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectr.* **5,** 976–989.
8. Eng, J. K., Fischer, B., Grossman, J. and MacCoss, M. J. (2008) A fast SEQUEST cross correlation algorithm. *J. Proteome Res.* **7,** 4598–4602.
9. Perkins, D. N., Pappin, D. J. C., Creasy, D. M. and Cottrell, J. S. (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **20,** 3551–3567.
10. Craig, R. and Beavis, R. C. (2003) A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Commun. Mass Sp.* **17,** 2310–2316.
11. Kim, S., Mischerikow, N., Bandeira, N., Navarro, J. D., Wich, L., Mohammed, S., Heck, A. J., Pevzner, P. A. (2010) The generating function of CID, ETD, and CID/ETD pairs of tandem mass spectra: applications to database search. *Mol. Cell. Proteomics* **9,** 2840–2852.
12. Park, C. Y., Klammer, A. A., Käll, L., MacCoss, M. P. and Noble, W. S. (2008) Rapid and accurate peptide identification from tandem mass spectra. *J. Proteome Res.* **7,** 3022–3027.

13. Šidák, Z. K. (1967) Rectangular confidence regions for the means of multivariate normal distributions. *J. Am. Stat. Assoc.* **62,** 626–33.
14. Käll, L., Storey, J. D., MacCoss, M. J. and Noble, W. S. (2008) Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J. Proteome Res.* **7,** 29–34.
15. Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J Roy. Stat. Soc. B* **57,** 289–300.
16. Hoopmann, M. R., Merrihew, G. E., von Haller, P. D. and MacCoss, M. J. (2009) Post analysis data acquisition for the iterative MS/MS sampling of proteomics mixtures. *J. Proteome Res.* **8,** 1870–1875.
17. Hsieh, E., Hoopmann, M., MacLean, B. and Maccoss, M. (2010) Comparison of database search strategies for high precursor mass accuracy MS/MS data. *J. Proteome Res.* **9,** 1138–1143.
18. Kline, K. G., Frewen, B., Bristow, M. R., MacCoss, M. J. and Wu, C. C. (2008) High quality catalog of proteotypic peptides from human heart. *J. Proteome Res.* **7,** 5055–5061.
19. Granholm, V., Kim, S., Navarro, J. C., Sjũlund, E., Smith, R. D., and Käll L. (2014) Fast and accurate database searches with MS-GF+Percolator. *J. Proteome Res.* **13,** 890–897.
20. Jeong, K., Kim, S. and Bandeira, N. (2012) False discovery rates in spectral identification. *BMC Bioinformatics* **13,** S2.
21. Granholm, V., Navarro, J. F., Noble, W. S. and Käll, L. (2013) Determining the calibration of confidence estimation procedures for unique peptides in shotgun proteomics. *J. Proteomics* **80,** 123–131.
22. Zhang, N., Li, X. J., Ye, M., Pan, S., Schwikowski, B., and Aebersold, R. (2005) ProbIDtree: an automated software program capable of identifying multiple peptides from a single collision-induced dissociation spectrum collected by a tandem mass spectrometer. *Proteomics* **5,** 4096–4106.
23. Wang, J., Pewithaccuterez-Santiago, J., Katz, J. E., Mallick, P., and Bandeira, N. (2010) Peptide identification from mixture tandem mass spectra. *Molecular and Cellular Proteomics* **9,** 1476–1485.
24. Alves, G., Ogurtsov, A. Y. and Yu, Y. K. (2010) RAId_aPS: MS/MS analysis with multiple scoring functions and spectrum-specific statistics. *PLoS ONE* **5,** e15438.